

# **УМ ПРОГРАММИСТА КАК ПОНЯТЬ И ОСМЫСЛИТЬ ЛЮБОЙ КОД**

**Фелин Херманс**

Предисловие Джона Скита

Санкт-Петербург

«БХВ-Петербург»

2023

УДК 004.4  
ББК 32.973.26-02  
Х39

**Херманс Ф.**

Х39 Ум программиста. Как понять и осмыслить любой код: Пер. с англ. — СПб.: БХВ-Петербург, 2023. — 272 с.: ил.

ISBN 978-5-9775-1176-6

Книга освещает практические основы когнитивистики для программистов. Основные темы: осмысление и развитие чужого и собственного кода, изучение новых языков программирования, мнемонические приемы для программистов, поддержка кода в читаемом состоянии. Объяснено, как снижать когнитивную нагрузку при работе программиста, как делать код логичным и понятным для себя и коллег. Рассмотрены приемы именования функций, классов и переменных, подходы к ведению репозитория, совместной разработке и доработке кода.

*Для программистов и других IT-специалистов*

УДК 004.4  
ББК 32.973.26-02

**Группа подготовки издания:**

Руководитель проекта	<i>Олег Сивченко</i>
Зав. редакцией	<i>Людмила Гауль</i>
Перевод с английского	<i>Кристины Черниковой</i>
Компьютерная верстка	<i>Ольги Сергиенко</i>
Оформление обложки	<i>Зои Канторович</i>

Original English language edition published by Manning Publications.  
Copyright (c) 2021 by Manning Publications.  
Russian-language edition copyright (c) 2022 by BHV. All rights reserved.

Оригинальное издание на английском языке опубликовано Manning Publications.  
© 2021 Manning Publications.  
Издание на русском языке © 2022 ООО «БХВ». Все права защищены.

Подписано в печать 02.09.22.  
Формат 70×100<sup>1/8</sup>. Печать офсетная. Усл. печ. л. 21,93.  
Тираж 1000 экз. Заказ № 5329.  
"БХВ-Петербург", 191036, Санкт-Петербург, Гончарная ул., 20  
Отпечатано с готового оригинал-макета  
ООО "Принт-М", 142300, М.О., г. Чехов, ул. Полиграфистов, д. 1

ISBN 978-1-61729-867-7 (англ.)  
ISBN 978-5-9775-1176-6 (рус.)

© Manning Publications, 2021  
© Перевод на русский язык, оформление. ООО "БХВ-Петербург",  
ООО "БХВ", 2023

Предисловие .....	13
От автора .....	15
Благодарности.....	17
О книге .....	19
Структура книги.....	19
Дискуссионный форум liveBook .....	20
Об авторе.....	21
Об обложке.....	23
<b>ЧАСТЬ I. ОБ УЛУЧШЕНИИ НАВЫКОВ ЧТЕНИЯ КОДА .....</b>	<b>25</b>
<b>Глава 1. Определение вашего типа замешательства при кодировании .....</b>	<b>27</b>
1.1. Разные типы замешательства в коде .....	28
1.1.1. Первый тип замешательства — недостаток знаний.....	29
1.1.2. Второй тип замешательства — недостаток информации.....	29
1.1.3. Третий тип замешательства — недостаток вычислительной мощности .....	30
1.2. Различные когнитивные процессы, влияющие на процесс кодирования .....	31
1.2.1. Долговременная память и программирование .....	31
Программа на APL с точки зрения долговременной памяти.....	32
1.2.2. Кратковременная память и программирование .....	32
Программа на Java с точки зрения кратковременной памяти .....	32
1.2.3. Рабочая память и программирование .....	33
Программа на BASIC с точки зрения рабочей памяти.....	33
1.3. Совместная работа когнитивных процессов .....	34
1.3.1. Краткое описание того, как когнитивные процессы взаимодействуют друг с другом.....	34
1.3.2. Когнитивные процессы и программирование .....	35
Выводы .....	37
<b>Глава 2. Скорочтение кода.....</b>	<b>39</b>
2.1. Быстрое чтение кода.....	40
2.1.1. Что только что происходило в вашем мозге .....	41
2.1.2. Перепроверка воспроизведенного кода .....	42
Вторая попытка воспроизведения кода .....	43

2.1.3. Перепроверка воспроизведенного.....	44
2.1.4. Почему читать незнакомый код так сложно.....	44
2.2. Преодоление лимитов памяти .....	45
2.2.1. Сила чанков .....	45
Чанки кода .....	48
2.2.2. Опытные программисты запоминают код лучше начинающих программистов.....	48
2.3. Вы видите намного больше кода, чем можете прочитать.....	49
2.3.1. Иконическая память .....	50
Иконическая память и код.....	51
2.3.2. Это не то, что вы помните; это то, как вы запоминаете .....	51
Как написать код, который можно разделить на чанки .....	52
Используйте паттерны проектирования.....	53
Пишите комментарии.....	54
Оставляйте «маячки» .....	55
2.3.3. Применяйте чанки .....	58
Выводы .....	59
<b>Глава 3. Как быстро выучить синтаксис .....</b>	<b>61</b>
3.1. Советы по запоминанию синтаксиса .....	62
3.1.1. Отвлечение снижает производительность .....	62
3.2. Как быстро выучить синтаксис с использованием карточек .....	63
3.2.1. Когда использовать карточки .....	64
3.2.2. Расширяем набор карточек .....	64
3.2.3. Убираем ненужные карточки.....	64
3.3. Как не забывать информацию .....	65
3.3.1. Почему мы забываем.....	65
Иерархия и сеть .....	66
Кривая забывания.....	66
3.3.2. Интервальное повторение .....	67
3.4. Как запомнить синтаксис надолго.....	69
3.4.1. Два способа запоминания информации .....	69
Уровень хранения.....	69
Уровень воспроизведения .....	69
3.4.2. Просто увидеть недостаточно.....	70
3.4.3. Воспоминания укрепляют память .....	70
3.4.4. Укрепление памяти путем активного мышления.....	71
Схемы .....	72
Проработка для запоминания концепций программирования .....	73
Выводы .....	74
<b>Глава 4. Как читать сложный код .....</b>	<b>75</b>
4.1. Почему так тяжело понимать сложный код .....	76
4.1.1. Чем друг от друга отличаются рабочая память и кратковременная память .....	77
4.1.2. Типы когнитивной нагрузки и как они связаны с программированием .....	78
Внутренняя когнитивная нагрузка при чтении кода .....	78
Внешняя когнитивная нагрузка при чтении кода .....	79
4.2. Способы снижения когнитивной нагрузки.....	80
4.2.1. Рефакторинг .....	80

4.2.2. Замена незнакомых языковых конструкций.....	82
Лямбда-функции.....	82
Генератор списков.....	83
Тернарные операторы.....	84
4.2.3. Синонимизация — отличное дополнение к дидактическим карточкам.....	85
4.3. Вспомогательные средства при перегрузке рабочей памяти.....	85
4.3.1. Создание графа зависимостей.....	86
4.3.2. Использование таблицы состояний.....	88
4.3.3. Сочетание графов зависимостей и таблиц состояний.....	91
Выводы.....	93

## **ЧАСТЬ II. ПРОДОЛЖАЕМ ДУМАТЬ О КОДЕ..... 95**

### **Глава 5. Совершенствуем навыки углубленного понимания кода ..... 97**

5.1. Роли переменных.....	98
5.1.1. Разные переменные выполняют разные действия.....	98
5.1.2. Одиннадцать ролей, охватывающие почти все переменные.....	99
5.2. Роли и принципы.....	101
5.2.1. Польза ролей.....	102
Практические советы по работе с ролями переменных.....	103
5.2.2. Венгерская нотация.....	104
Системная и прикладная венгерские нотации.....	104
5.3. Углубленное понимание программ.....	106
5.3.1. Понимание текста и понимание плана.....	106
5.3.2. Этапы понимания программы.....	106
Применение этапов углубленного понимания.....	107
5.4. Чтение кода как обычного текста.....	109
5.4.1. Что происходит в мозге при чтении кода.....	110
Поля Бродмана.....	110
Показания фМРТ.....	111
5.4.2. Если вы можете выучить французский, то сможете выучить и Python.....	111
Как люди читают код.....	113
Перед тем как читать код, программисты сканируют его.....	114
Начинающие и опытные программисты читают код по-разному.....	114
5.5. Стратегии понимания текста, которые можно применить к коду.....	115
5.5.1. Активация пассивных знаний.....	116
5.5.2. Наблюдение.....	116
5.5.3. Определение важности разных строк кода.....	117
5.5.4. Предположения о значении имен переменных.....	118
5.5.5. Визуализация.....	119
Таблица операций.....	119
5.5.6. Постановка вопросов.....	120
5.5.7. Резюмирование кода.....	121
Выводы.....	122

### **Глава 6. Совершенствуем навыки решения задач программирования ..... 123**

6.1. Использование моделей для размышлений о коде.....	124
6.1.1. Преимущества использования моделей.....	124
Не все модели одинаково полезны.....	125

6.2. Ментальные модели .....	126
6.2.1. Подробное исследование ментальных моделей .....	128
6.2.2. Изучение новых ментальных моделей .....	129
6.2.3. Как эффективно использовать ментальные модели во время размышлений о коде .....	130
Ментальные модели в рабочей памяти.....	130
Точные модели работают лучше.....	131
Создание ментальных моделей исходной программы в рабочей памяти.....	131
Ментальные модели в долговременной памяти .....	132
Создание ментальных моделей исходной программы в долговременной памяти.....	133
Ментальные модели, одновременно хранящиеся в долговременной и рабочей памяти.....	134
6.3. Условные машины .....	135
6.3.1. Что такое условная машина .....	135
6.3.2. Примеры условных машин .....	136
6.3.3. Разные уровни условных машин .....	137
6.4. Условные машины и язык.....	138
6.4.1. Расширяем набор условных машин .....	139
6.4.2. Разные условные машины могут создать взаимно конфликтующие ментальные модели.....	140
6.5. Условные машины и схемы .....	141
6.5.1. Почему схема важна .....	141
6.5.2. Являются ли условные машины семантическими .....	142
Выводы .....	142
<b>Глава 7. Заблуждения .....</b>	<b>143</b>
7.1. Почему второй язык программирования выучить намного проще, чем первый .....	144
7.1.1. Как увеличить шанс воспользоваться знаниями по программированию.....	146
7.1.2. Разные виды трансференции.....	147
Осознанная и неосознанная трансференция .....	147
Близкая и дальняя трансференция .....	147
7.1.3. Знания: добро или зло?.....	148
7.1.4. Сложности трансференции .....	149
7.2. Заблуждения. Ошибки в мышлении.....	150
7.2.1. Исправление заблуждений путем концептуальных замен.....	152
7.2.2. Подавление заблуждений .....	152
7.2.3. Заблуждения о языках программирования .....	153
7.2.4. Предотвращение заблуждений при изучении нового языка программирования.....	155
7.2.5. Выявление заблуждений в новой базе кода.....	156
Выводы .....	157
<b>ЧАСТЬ III. О ХОРОШЕМ КОДЕ .....</b>	<b>159</b>
<b>Глава 8. Совершенствуем навыки присваивания имен.....</b>	<b>161</b>
8.1. Почему присваивание имен так важно .....	162
8.1.1. Почему присваивание имени так важно .....	162
Имена составляют существенную часть кодовой базы.....	163

Имена играют роль в обзорах кода.....	163
Имена — это самая удобная форма документации .....	163
Имена могут служить маячками .....	163
8.1.2. Разные точки зрения на присваивание имен .....	163
Хорошее имя можно определить синтаксически .....	164
Имена во всей базе кода должны быть единообразны.....	165
8.1.3. Важно грамотно подбирать имена .....	166
Заключения о практике присваивания имен .....	167
8.2. Когнитивные аспекты присваивания имен.....	167
8.2.1. Форматирование имен поддерживает кратковременную память .....	168
8.2.2. Понятные имена лучше закрепляются в долговременной памяти .....	169
8.2.3. Полезная информация в именах переменных .....	170
8.2.4. Когда стоит оценивать качество имен .....	171
8.3. Какие типы имен проще всего понимать.....	172
8.3.1. Использовать аббревиатуры или нет?.....	172
Однобуквенные имена переменных.....	173
8.3.2. Змеиный или верблюжий регистры?.....	176
8.4. Влияние имен на ошибки кода .....	177
8.4.1. В коде с некачественными именами больше ошибок.....	177
8.5. Как выбирать хорошие имена.....	178
8.5.1. Шаблоны имен .....	178
8.5.2. Трехступенчатая модель Фейтельсона для хороших имен переменных.....	180
Трехступенчатая модель во всех деталях.....	181
Успех трехступенчатой модели Фейтельсона.....	181
Выводы .....	182

## Глава 9. Боремся с плохим кодом и когнитивной нагрузкой.

<b>Две концепции .....</b>	<b>183</b>
9.1. Почему код с запахами кода создает большую когнитивную нагрузку.....	184
9.1.1. Краткая информации о запахах кода.....	184
Запахи кода на уровне метода .....	186
Запахи кода на уровне класса .....	186
Запахи кода на уровне базы кода .....	187
Влияние запахов кода .....	187
9.1.2. Как запахи кода вредят мышлению.....	188
«Длинный список параметров», сложные «Операторы переключения» — перегрузка рабочей памяти.....	188
«Всемогущий класс», «Длинный метод» — невозможно эффективно разбить код на чанки .....	189
«Клоны кода» — невозможно правильно разбить код на чанки .....	189
9.2. Зависимость когнитивной нагрузки от плохих имен.....	190
9.2.1. Лингвистические антипаттерны проектирования .....	190
9.2.2. Измерение когнитивной нагрузки .....	191
Шкала Пааса для когнитивной нагрузки .....	192
Измерение нагрузки по глазам .....	193
Измерение нагрузки по коже.....	193
Измерение нагрузки по мозгу .....	194
Запись биотоков мозга .....	194
Функциональная fNIRS-томография и программирование .....	195

9.2.3. Лингвистические антипаттерны и когнитивная нагрузка.....	195
9.2.4. Почему лингвистические антипаттерны вызывают замешательство.....	196
Выводы.....	197
<b>Глава 10. Совершенствуем навыки решения сложных задач.....</b>	<b>199</b>
10.1. Что такое решение задач.....	200
10.1.1. Элементы решения задач.....	200
10.1.2. Пространство состояний.....	200
10.2. Какую роль при решении задач программирования играет долговременная память.....	201
10.2.1. Решение задачи — это отдельный когнитивный процесс?.....	201
При решении задач вы используете долговременную память.....	202
Вашему мозгу проще решить знакомые задачи.....	202
10.2.2. Как научить долговременную память решать задачи.....	203
10.2.3. Два вида памяти, наиболее существенные при решении задачи.....	203
Какие виды памяти играют роль при решении задач.....	204
Потеря знаний или навыков.....	205
10.3. Автоматизация: создание имплицитной памяти.....	206
10.3.1. Имплицитная память с течением времени.....	207
Когнитивный этап.....	208
Ассоциативный этап.....	208
Автономный этап.....	209
10.3.2. Почему автоматизация помогает программировать быстрее.....	210
10.3.3. Улучшение имплицитной памяти.....	211
10.4. Обучение на основе кода и его объяснения.....	212
10.4.1. Новый вид когнитивной нагрузки: соответствующая нагрузка.....	213
10.4.2. Примеры с решением на практике.....	215
Работайте вместе с коллегой.....	215
Используйте GitHub.....	215
Читайте книги или блоги об исходном коде.....	216
Выводы.....	216
<b>ЧАСТЬ IV. О СОВМЕСТНОЙ РАБОТЕ НАД КОДОМ.....</b>	<b>217</b>
<b>Глава 11. Процесс написания кода.....</b>	<b>219</b>
11.1. Различные активности, выполняемые во время программирования.....	220
11.1.1. Поиск.....	220
11.1.2. Осмысление.....	221
11.1.3. Переписывание.....	221
11.1.4. Нарращивание.....	222
11.1.5. Исследование.....	222
11.1.6. А как же отладка?.....	223
11.2. Программист отвлекся.....	223
11.2.1. Задачи программирования нуждаются в «разогреве».....	224
11.2.2. Что происходит после отвлечения.....	225
11.2.3. Как подготовиться к отвлечению.....	225
Сохраняйте воображаемую модель.....	225
Помогите своей проспективной памяти.....	226
Определитесь с промежуточными целями.....	228



11.2.4. Когда отвлекать программиста .....	228
11.2.5. Пара слов о многозадачности.....	230
Многозадачность и автоматизация.....	230
Исследования многозадачности.....	231
Выводы .....	231
<b>Глава 12. Проектирование и усовершенствование больших систем.....</b>	<b>233</b>
12.1. Проверка свойств базы кода .....	234
12.1.1. Когнитивные измерения .....	234
Подверженность ошибкам .....	235
Согласованность .....	236
Размытость .....	236
Скрытые зависимости.....	237
Преждевременная фиксация решения.....	238
Вязкость.....	238
Поэтапное оценивание.....	239
Выразительность ролей .....	239
Близость соответствия .....	240
Трудность мыслительных операций.....	241
Вторичные обозначения .....	242
Градиент абстракции .....	242
Наглядность.....	243
12.1.2. Использование когнитивных измерений базы кода для улучшения базы кода.....	243
12.1.3. Проектные маневры и их плюсы и минусы.....	244
Подверженность ошибкам и вязкость.....	244
Преждевременная фиксация решения и поэтапное оценивание против подверженности ошибкам .....	245
Выразительность ролей и размытость .....	245
12.2. Измерения и активности .....	245
12.2.1. Влияние измерений на разные активности.....	245
Поиск.....	245
Осмысление .....	246
Переписывание.....	246
Наращивание .....	246
Исследование.....	247
12.2.2. Изменение базы кода под ожидаемые активности .....	247
Выводы .....	247
<b>Глава 13. Как ввести новых программистов в курс дела .....</b>	<b>249</b>
13.1. Проблемы процесса адаптации.....	249
13.2. Различия между профессионалами и новичками.....	251
13.2.1. Поведение новичка более подробно .....	251
Оригинальная концепция Пиаже .....	251
Концепция неопиажизма для программирования .....	252
При изучении новой информации вы можете временно забывать некоторые вещи.....	255
13.2.2. Разница между вещественным и абстрактным видением концепций.....	255

13.3. Активности для улучшения процесса адаптации.....	258
13.3.1. Ограничение заданий до одной активности.....	258
13.3.2. Поддержка памяти новичка.....	259
Поддержка долговременной памяти: объяснение релевантной информации.....	259
Поддержка кратковременной памяти: ставьте небольшие конкретные задачи.....	260
Поддержка рабочей памяти: используйте диаграммы.....	261
13.3.3. Совместное чтение кода.....	261
Активация.....	262
Определение важности.....	262
Постановка предположений.....	262
Наблюдение.....	262
Визуализация.....	263
Постановка вопросов.....	263
Резюмирование.....	263
Выводы.....	263
<b>Эпилог. Пара слов перед прощанием.....</b>	<b>265</b>
<b>Предметный указатель.....</b>	<b>267</b>